FORTINET NSE ACADEMY XPERTS

Public Cloud

Deploying the Security Fabric as Infrastructure as Code in Microsoft Azure

Joeri Van Hoof

@jvhoof

What is Infrastructure as Code all about?

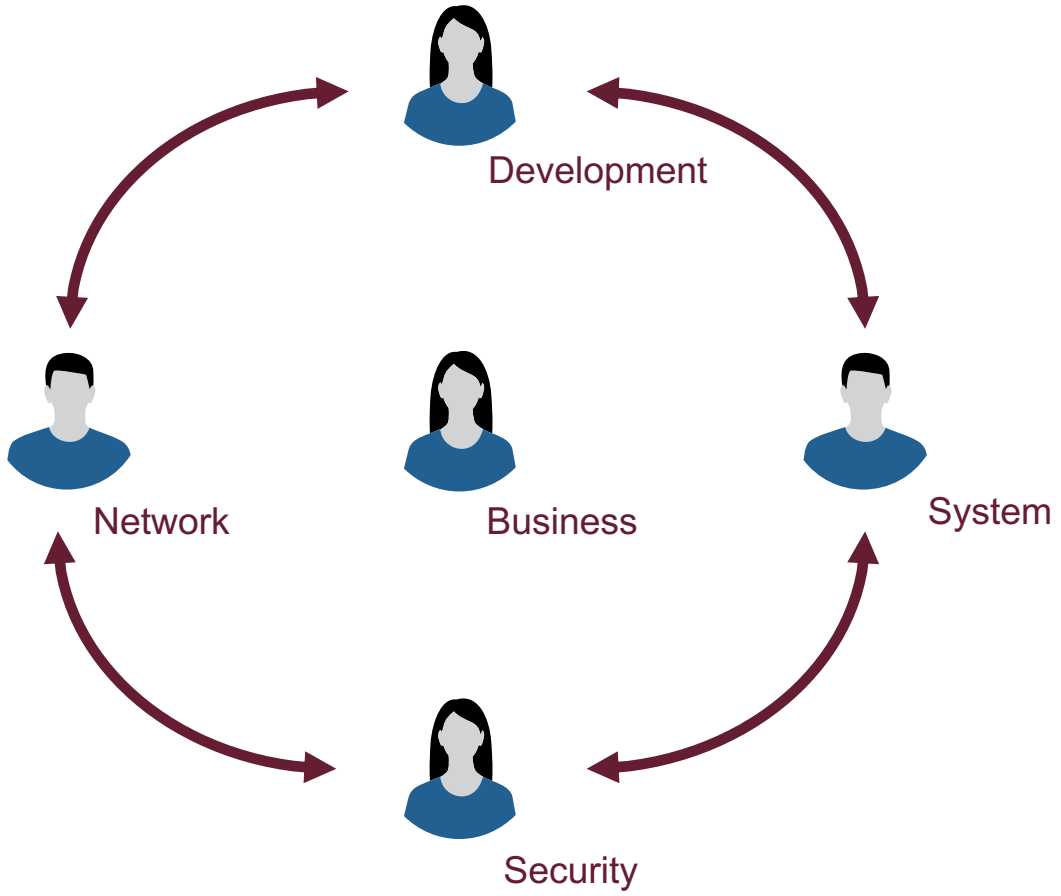
A little story to start…

Asterix and Obelix - The 12 tasks

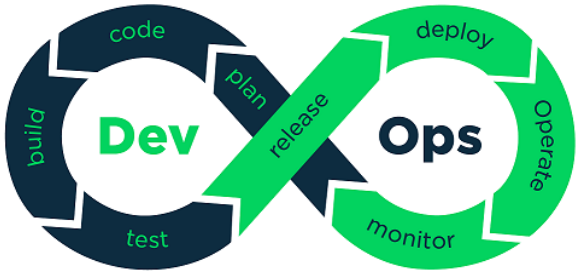
https://www.youtube.com/watch?v=JOhRhq6Pr6g
(watch around minute 40 and on)

FORTINET NSE ACADEMY XPERTS

A little story to start…

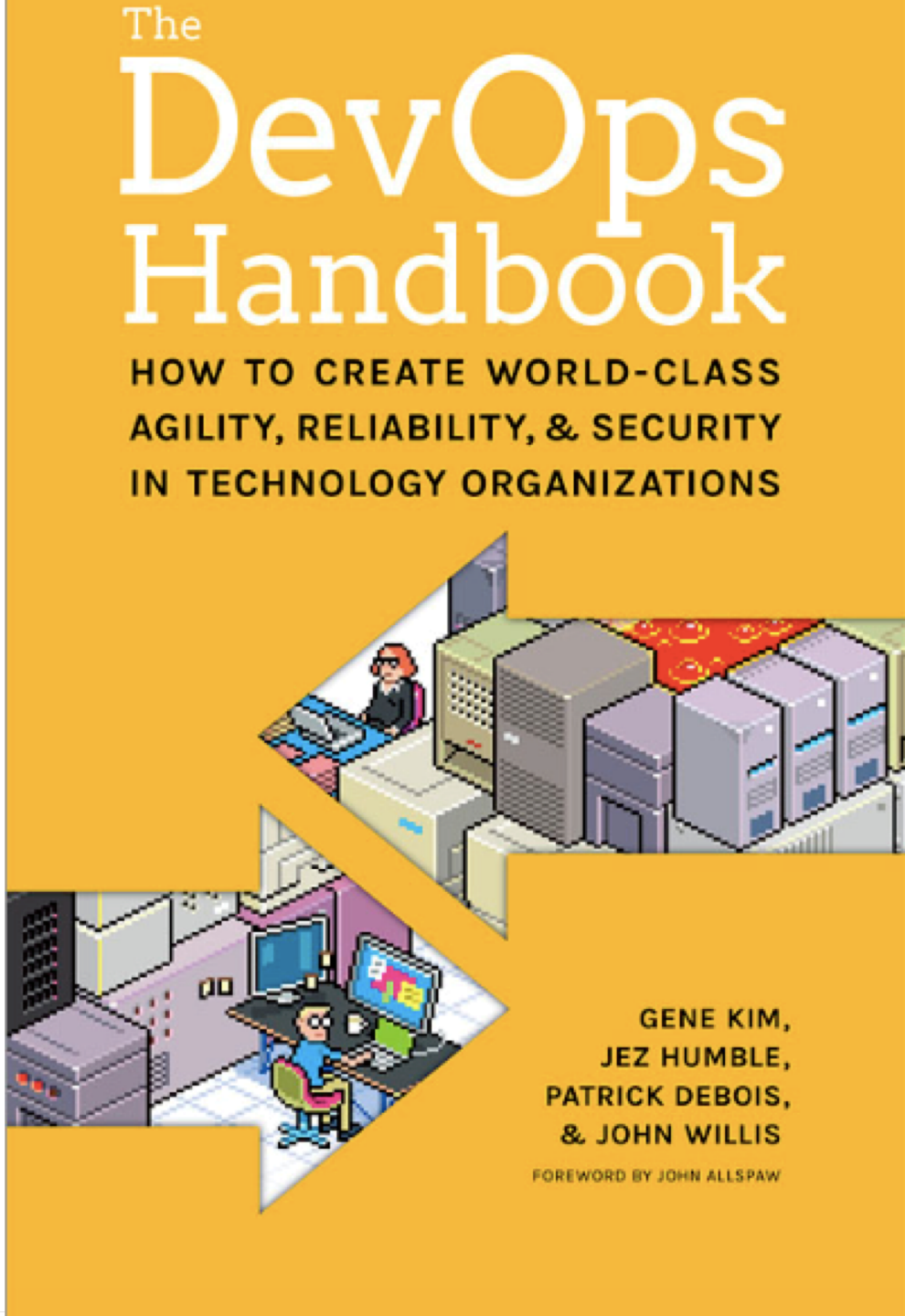


DevOps


code
plan
build
test
release
deploy
operate
monitor
Dev
Ops

What is DevOps?

- “DevOps and its resulting technology, architectural, and cultural practices represent a convergence of many philosophical and management movements….
- …DevOps is the outcome of applying the most trusted principles from the domain of physical manufacturing and leadership to the IT value steam DevOps relies on bodies of knowledge from **Lean**, **Theory of Constraints**, the **Toyota Production System**, **reliance engineering**, **learning organizations**, **safety culture**, **human factors**, and many others.”
  - ~ *The DevOps Handbook*



The Phoenix Project

"*The Phoenix Project*" describes the underpinning principles that all the DevOps patterns can be derived from '**The Three Ways**'.

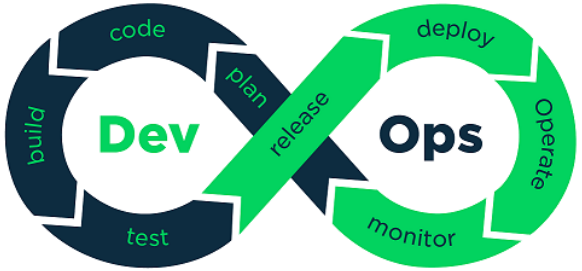
- **First Way**: left-to-right flow of work from Development to IT Operations to the customer.
- **Second Way**: constant flow of fast feedback from right-to-left at all stages of the value stream,
- **Third Way**: creating a culture that fosters continual experimentation, identify repetition and practice.

DevOps

Development vs. Operation

How can we make this cooperation smoother?

Started with Devops Days in Ghent 2009



Infrastructure as Code

Infrastructure as Code

What?

The essence of IaC is to treat the **configuration** and **definition** of **systems** just **like** writing **software**

Infrastructure as Code

Why?

- Repeatability
- Speed
- Accuracy
- Testability
- Agility

Infrastructure as Code

Advantages

- Not just for cloud, software defined data center
- Embedded documentation
- Source control
- Flexible build process
- Removes or reduces configuration drift

Infrastructure as Code

People

- Required to be successful
- A way of life
- Connects into the DevOps movement
- IaC / DevOps movement
- Grassroots vs top down

Process

- Simplicity
- Modular
- Flexible
- Versioning

Products

- Terraform, Ansible, Salt, Chef, Puppet
- ARM templates
- Powershell / Bash
- Azure Quickstart templates
- Visual Studio Code
- CI/CD pipeline tools like:
  - Azure Devops
  - Jenkins
  - Circle CI
  - …

Infrastructure as Code

People: your first steps

- Find something small to automate
  - Fail fast, fail often!
- Set your and the organisations expectations
  - Don’t just fall for buzzwords!
- Verify that is works
  - Did this help in our deployment efforts?
- Engage your peers
  - Share your experience
- Rinse and repeat to refine the process

Infrastructure as Code

Process

GitHub

Jenkins

Azure DevOps

puppet

ANSIBLE

HashiCorp Terraform

Microsoft OMS

Grafana

elastic + logstash + kibana

CI
Continuous Integration

CD
Continuous Deployment

FORTINET NSE ACADEMY XPERTS

Before the products

some extra concepts that are important…

Configuration management

"Configuration management is the process of **standardizing resource configurations** and **enforcing** their **state** across IT infrastructure in an automated yet agile manner."

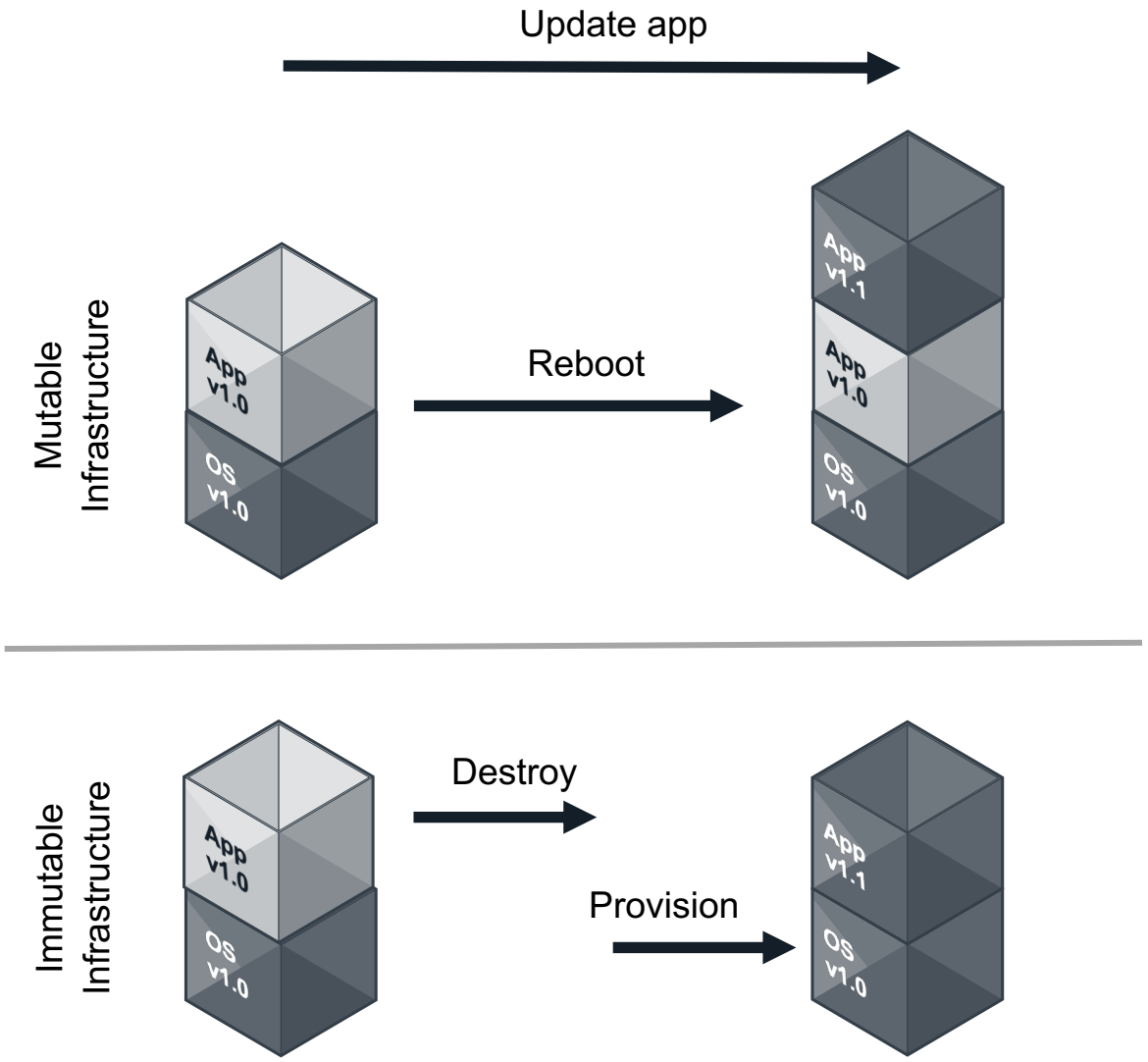
Puppet Labs

Idempotence

Idempotence is the property that a deployment command always sets the **target environment** into the **same configuration, regardless** of the environment's **starting state**.

Mutable vs. Immutable

- Immutable means deploy will never change
- Image regenerated each time
- Reuse the image from dev to QA to Prod
- To scale spin up an new instance
- Reliability, predictability



FORTINET NSE ACADEMY XPERTS

Imperative vs Declarative

- Imperative (procedural): defines specific commands that need to be executed in the appropriate order to arrive at the end state

- Aka “the HOW”

```
# AZURE CLI:
# Create resource group

rg=“myResourceGroup”
Location=“westeurope”

echo "--> Creating $rg resource
az group create --location "$location" --name "$rg"
```

FORTINET NSE ACADEMY XPERTS

Imperative vs Declarative

- Declarative (functional): defines the desired state and the system executes what needs to happen to achieve that desired state

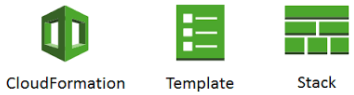
- Aka “the WHAT”

```
# Azure ARM Template
…
"resources": [
        {
                "type": "Microsoft.Resources/resourceGroups",
                "apiVersion": "2018-05-01",
                "location": "[parameters('rgLocation')]",
                "name": "[parameters('rgName')]",
                "properties": {}
        }
]
…
```

Products overview

Tool	Tool type	Infrastructure	Architecture	Approach	Language
puppet	Configuration Management	Mutable	Pull	Declarative	DSL + Ruby
CHEF	Configuration Management	Mutable	Pull	Declarative & Imperative	Ruby
ANSIBLE	Configuration Management	Mutable	Push	Declarative & Imperative	YAML
SALTSTACK	Configuration Management	Mutable	Push & Pull	Declarative & Imperative	YAML
HashiCorp Terraform	Provisioning	Immutable	Push	Declarative	HCL

Products overview

Tool	Tool type	Infrastructure	Architecture	Approach	Language
AWS CloudFormation Templates 	Provisioning	Immutable	Push	Declarative	JSON
ARM Templates 	Provisioning	Immutable	Push	Declarative	JSON or YAML
	Provisioning	Immutable	Push	Declarative	HCL

So many choices!

What do we see in the field?

- Azure: ARM Templates
- AWS: CFT templates
- Multi-Cloud: Terraform + Ansible

So many choices!

During the workshop?

- Microsoft Azure
- Azure DevOps
- Azure Resource Manager (ARM) Templates
- Terraform
- Ansible
- Azure Cloud Shell

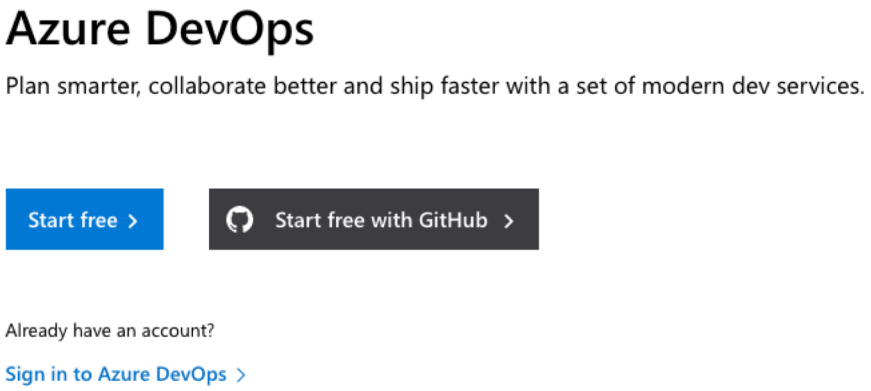
FORTINET NSE ACADEMY XPERTS

Infrastructure as Code

Our first baby steps

ARM template validation

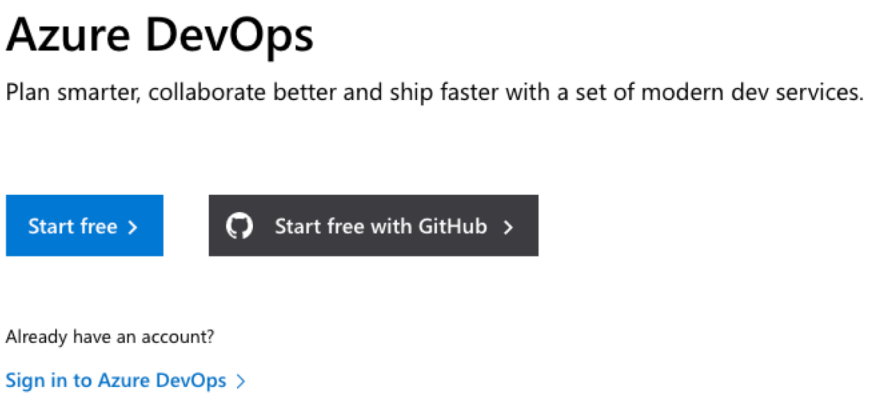
What if you are developing an ARM templates? Once you commit to git is this template correct? Can we automate the testing in a repeatable, consistent way?



ARM template validation

The ingredients

- Microsoft Azure account
- Microsoft Azure DevOps account (free):
  - https://azure.microsoft.com/en-in/services/devops/
  - https://github.com/jvhoof/xperts-academy-2019/
  - az vm image accept-terms --publisher fortinet --offer fortinet_fortigate-vm_v5 --plan fortinet_fg-vm
- Optional: Github account



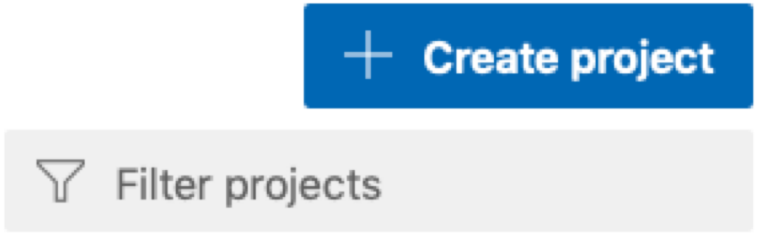
FORTINET NSE ACADEMY XPERTS

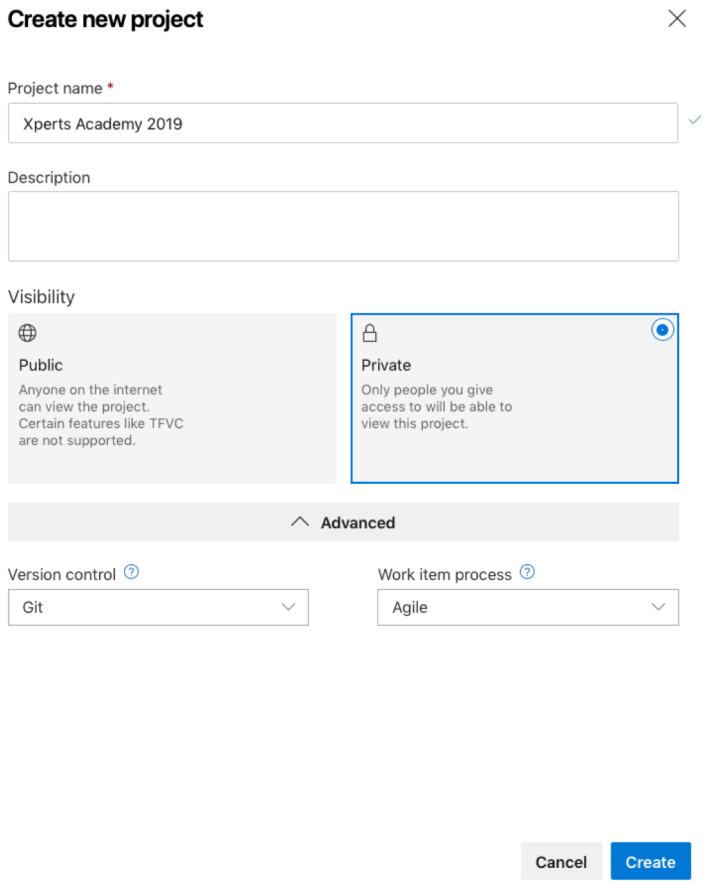
ARM template validation

Create a new project in the main organization page.

Xperts Academy 2019

Visibility can be public or private.

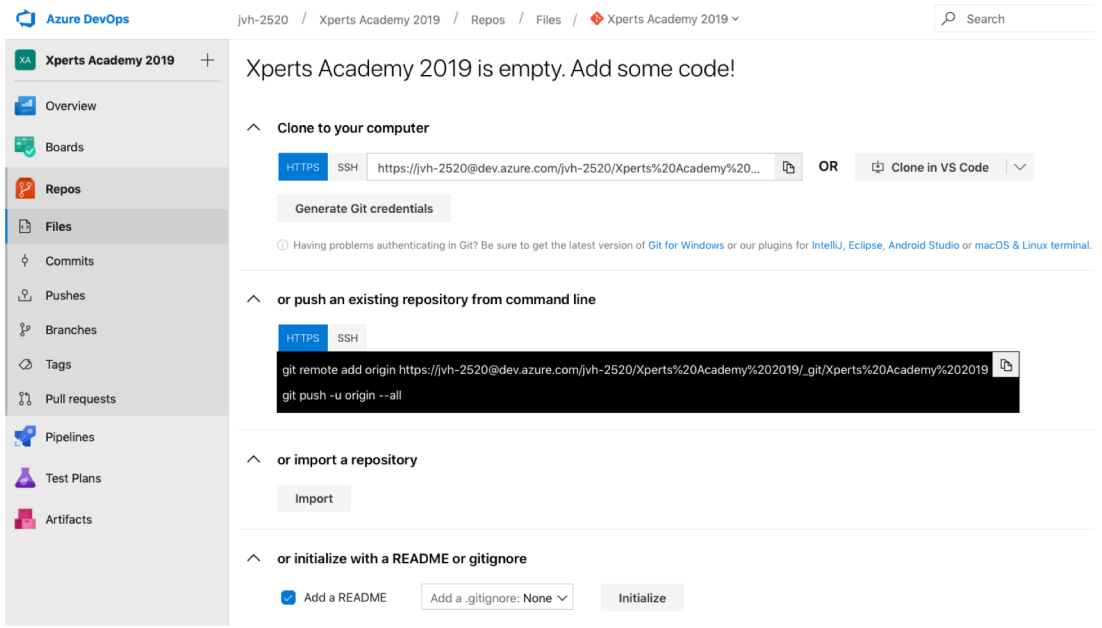




FORTINET NSE ACADEMY XPERTS

Import the ARM Template project

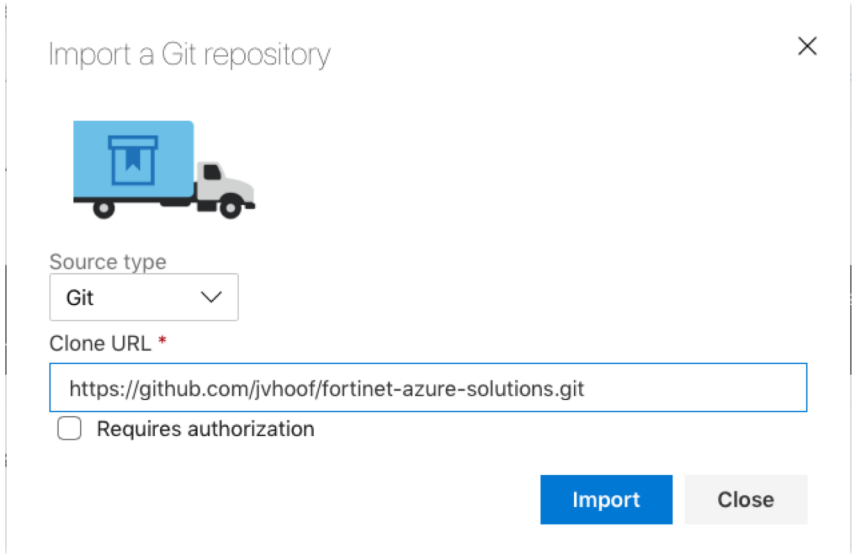
Go to Repos and select import

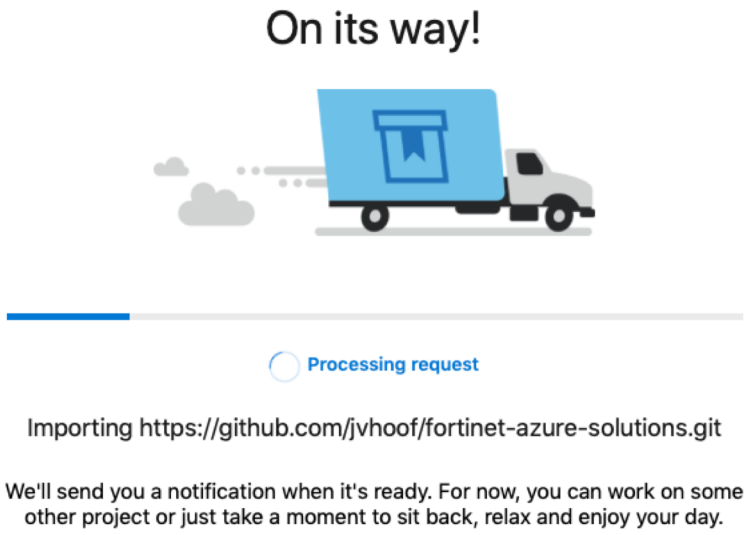


Import the ARM Template project

Import from the following Github project:

https://github.com/jvhoof/Fortinet-azure-solutions.git



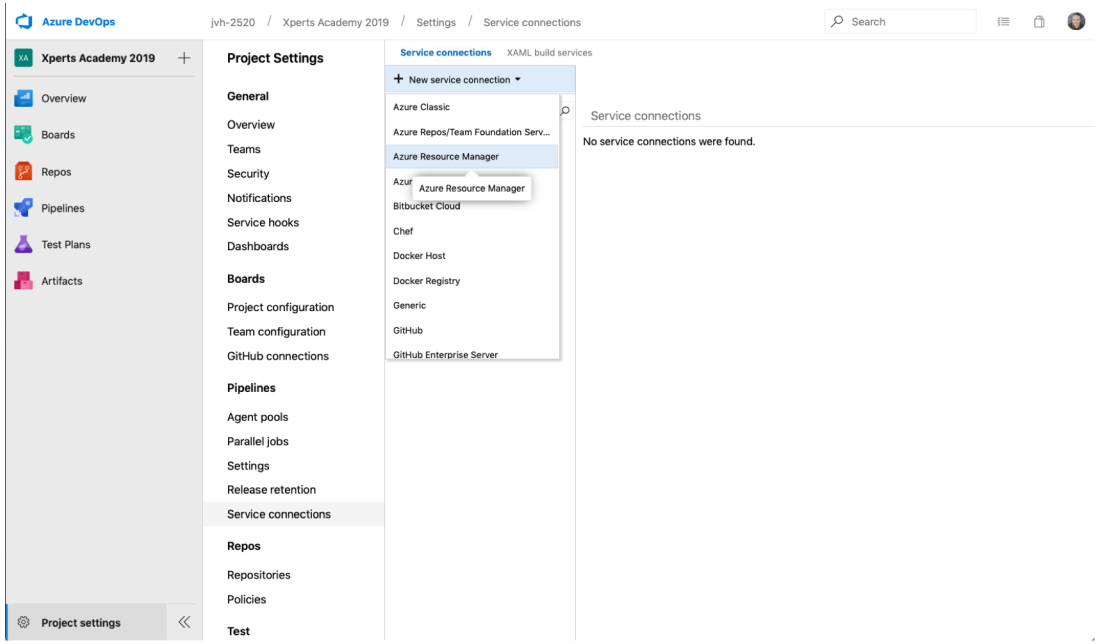


FORTINET NSE ACADEMY XPERTS

Connect Azure DevOps with Azure

Open the **Project settings** and look for **Service Connections**

Create a **New Service Connections** for Azure Resource Manager

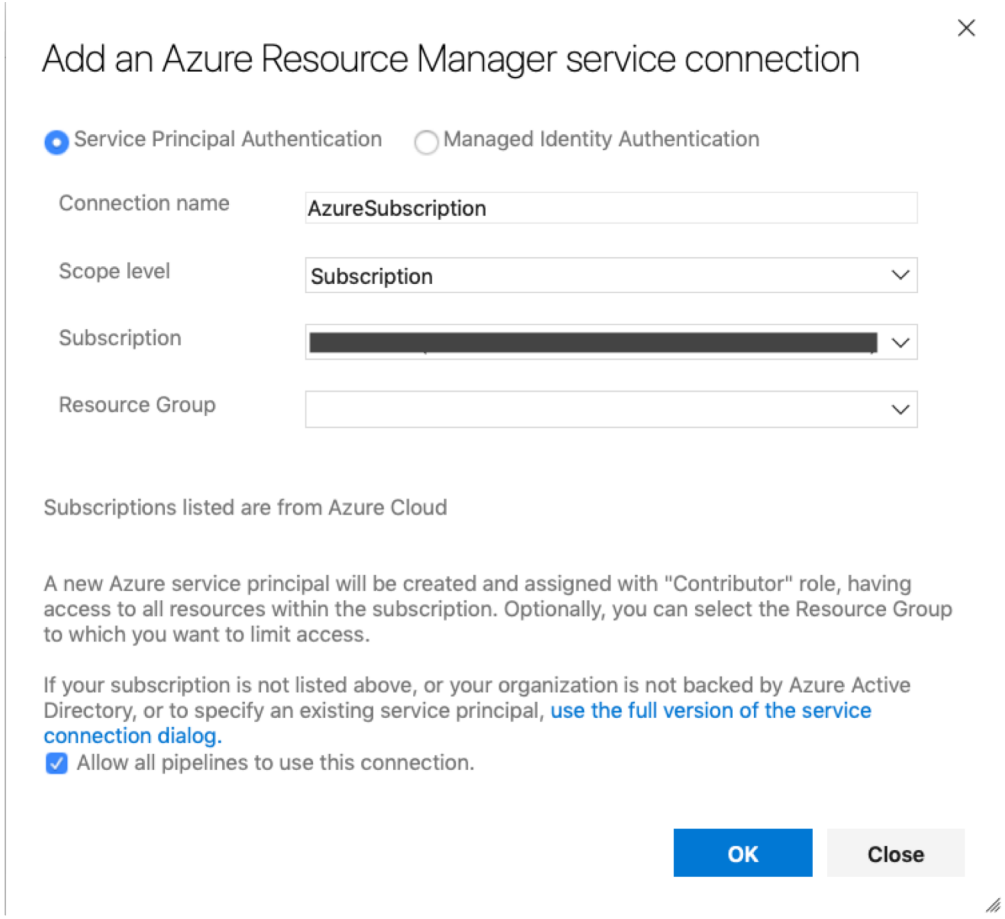


FORTINET NSE ACADEMY XPERTS

Connect Azure DevOps with Azure

Use the automatic way if you use the same credentials for Azure and Azure DevOps

Give the Connection the following name in one word and no quote's: "AzureSubscription"

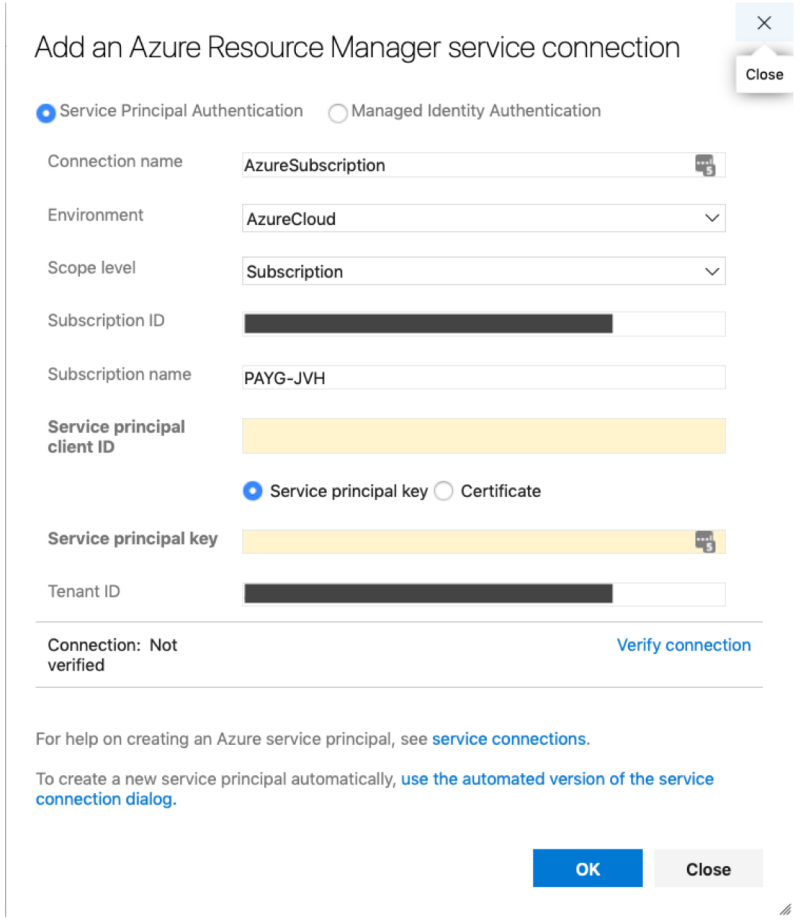


Connect Azure DevOps with Azure

Create a **service principal** or use **managed identity authentication** in case you have a different account

https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal

https://docs.microsoft.com/en-us/powershell/azure/create-azure-service-principal-azureps?view=azps-2.3.2

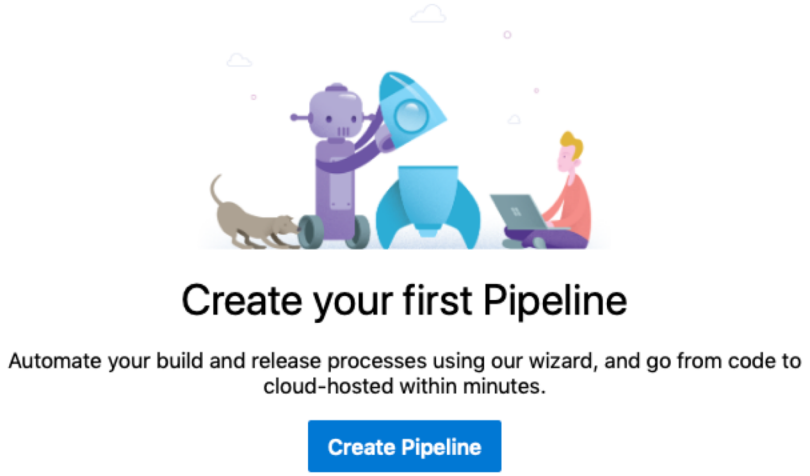


FORTINET NSE ACADEMY XPERTS

Create your first build pipeline

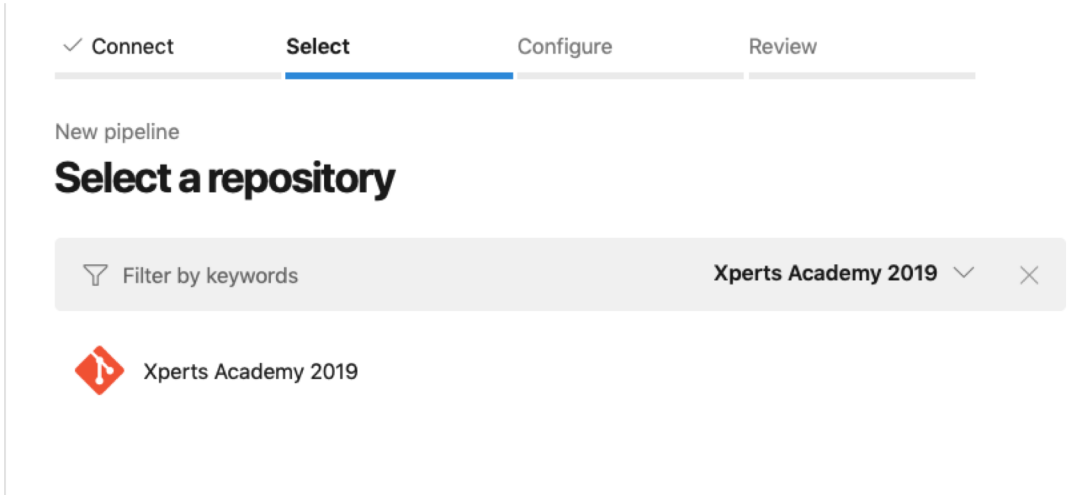
Open the **Pipeline** menu in the blade

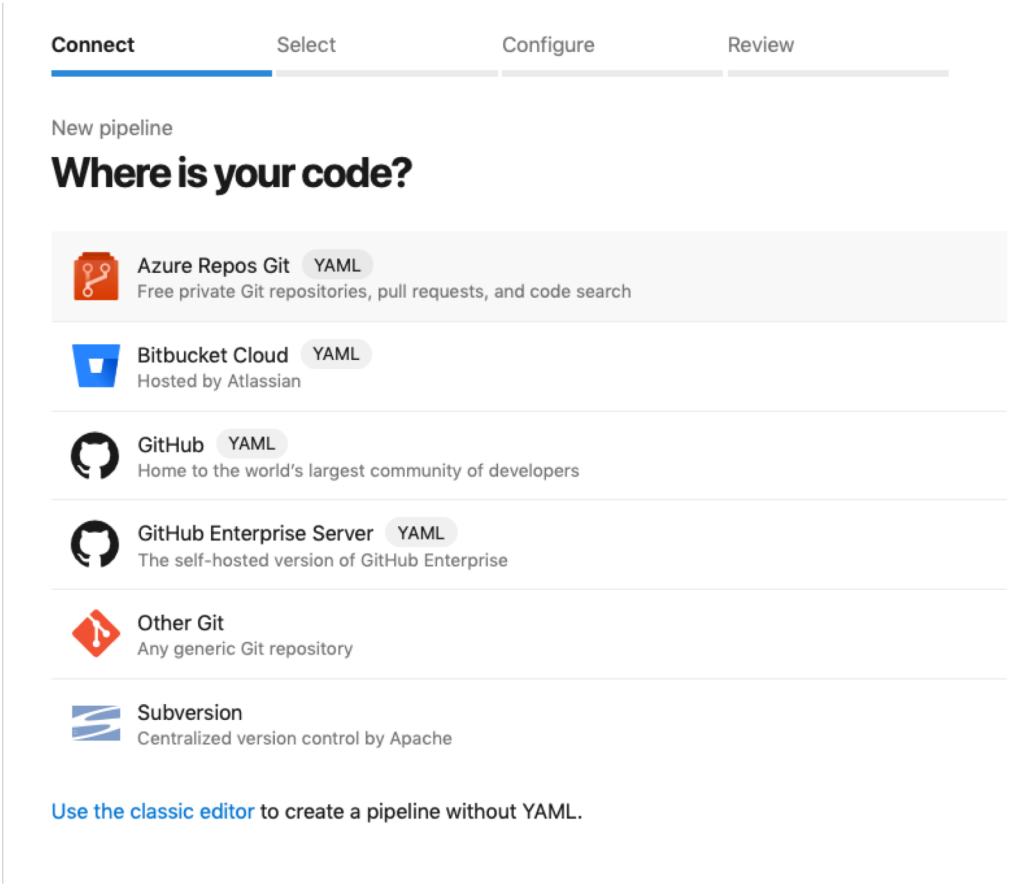
Click on the ‘Create Pipeline’ button



Create your first build pipeline

Select the Git repository you have the code imported to





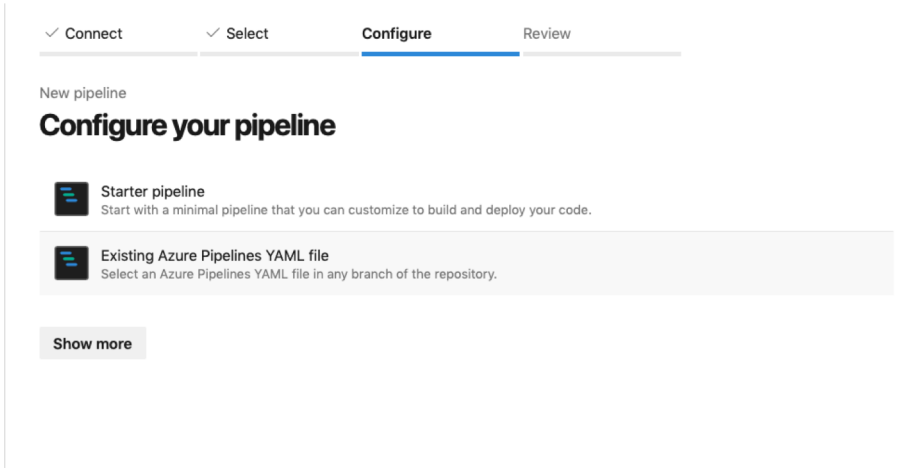
FORTINET NSE ACADEMY XPERTS

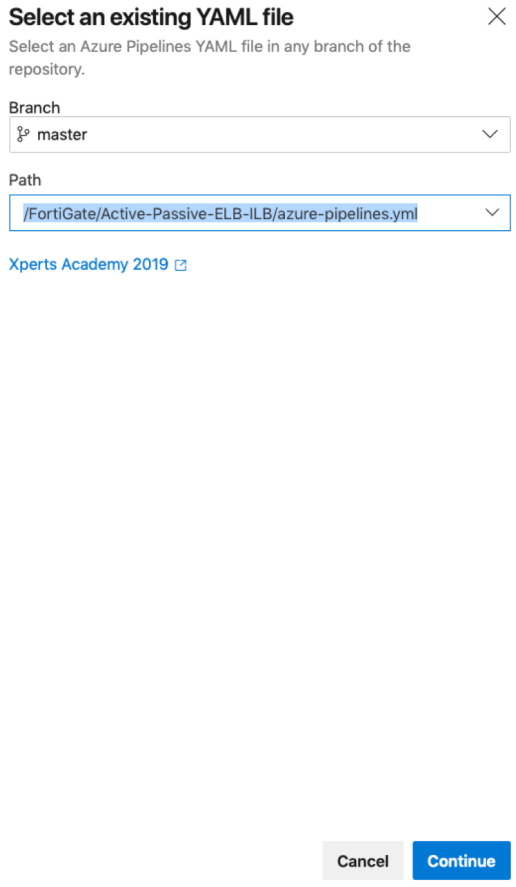
Create your first build pipeline

Azure Pipelines like most CI/CD pipelines have a configuration YAML file per project they can read in with the required steps

Select one of the three YAML pipelines Azure DevOps offers you

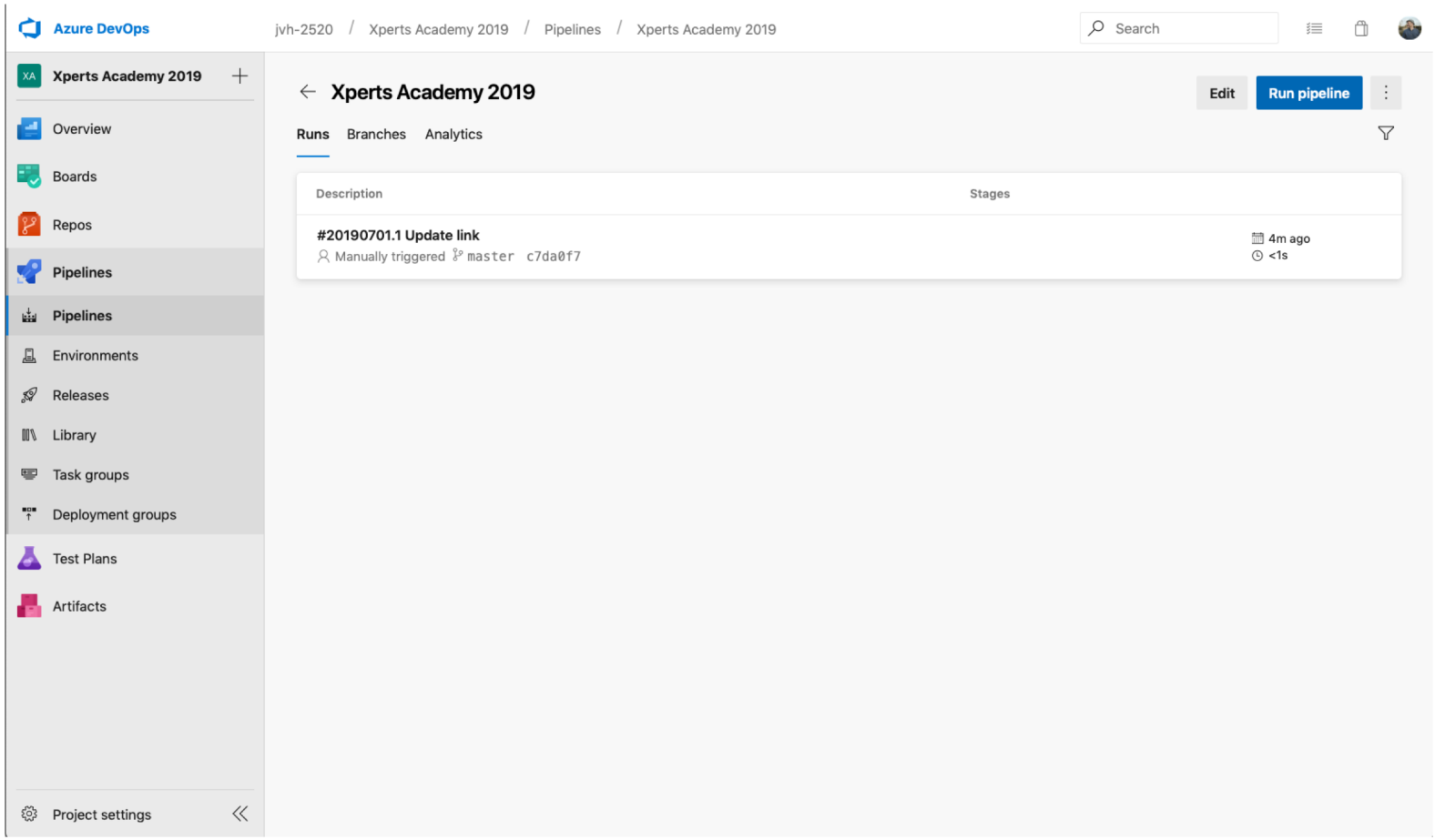
And run your pipeline!



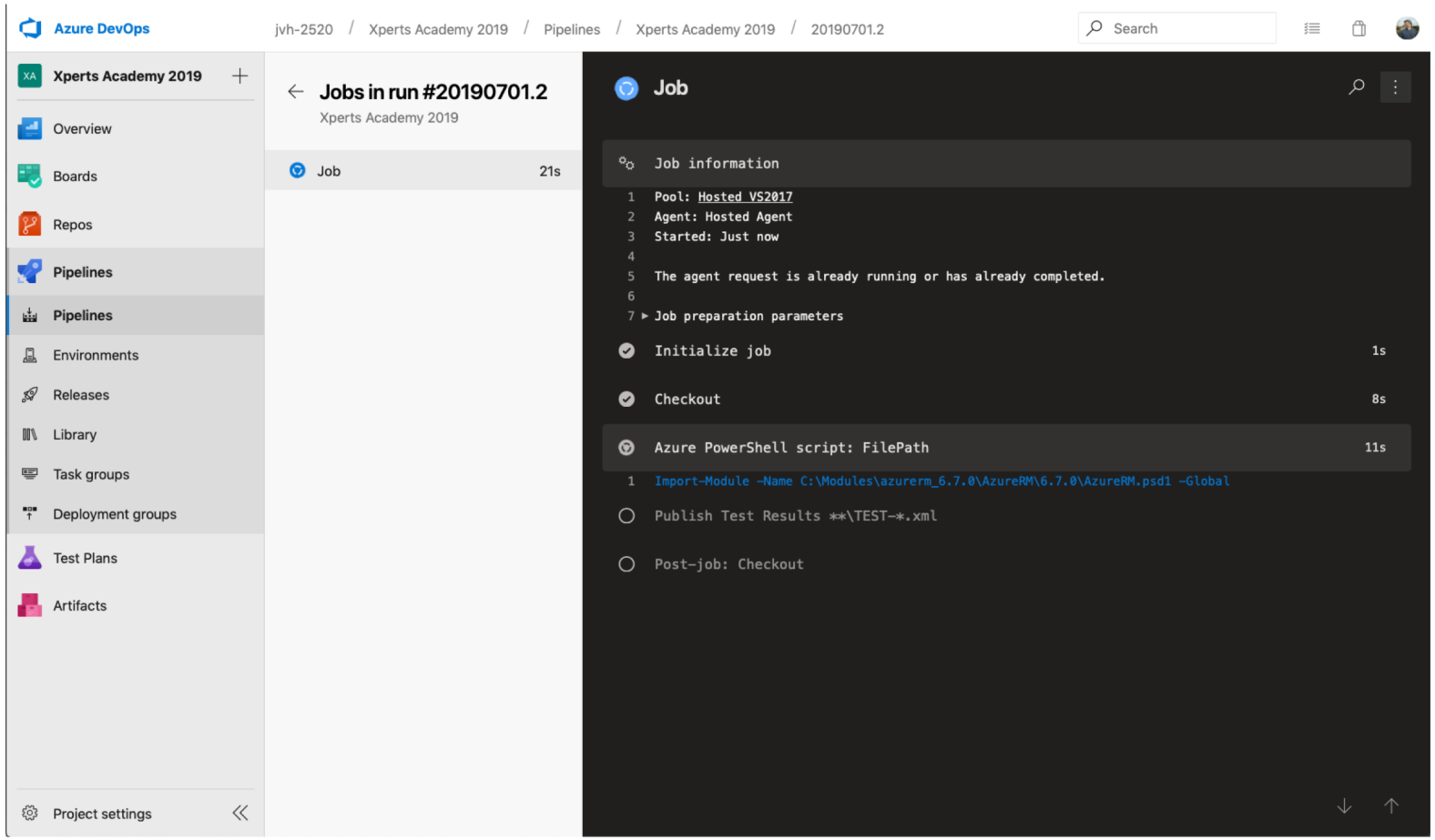


FORTINET NSE ACADEMY XPERTS

The result



The result



Demo and what's next

- Continuous integration
- Trigger
- Import code in Visual Studio Code
- Commit an update

Infrastructure as Code

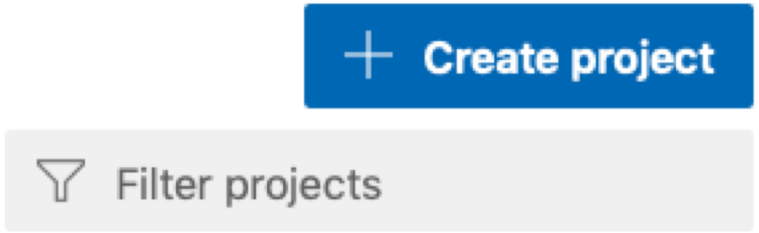
Step 2 : Build and release

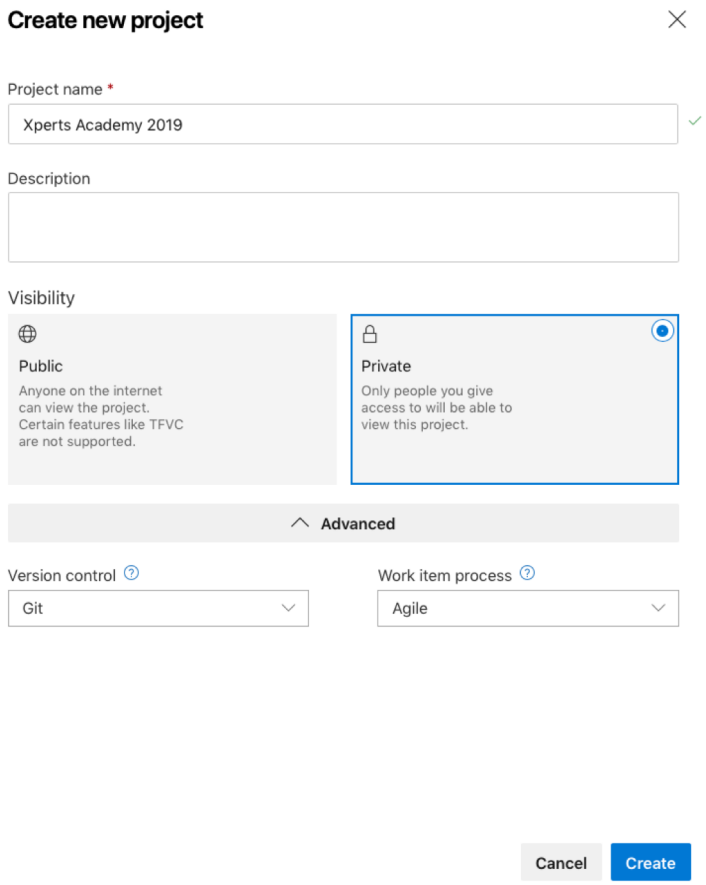
Build and release

Create a new project in the main organization page:

Xperts Academy 2019 Part 2

Visibility can be public or private.

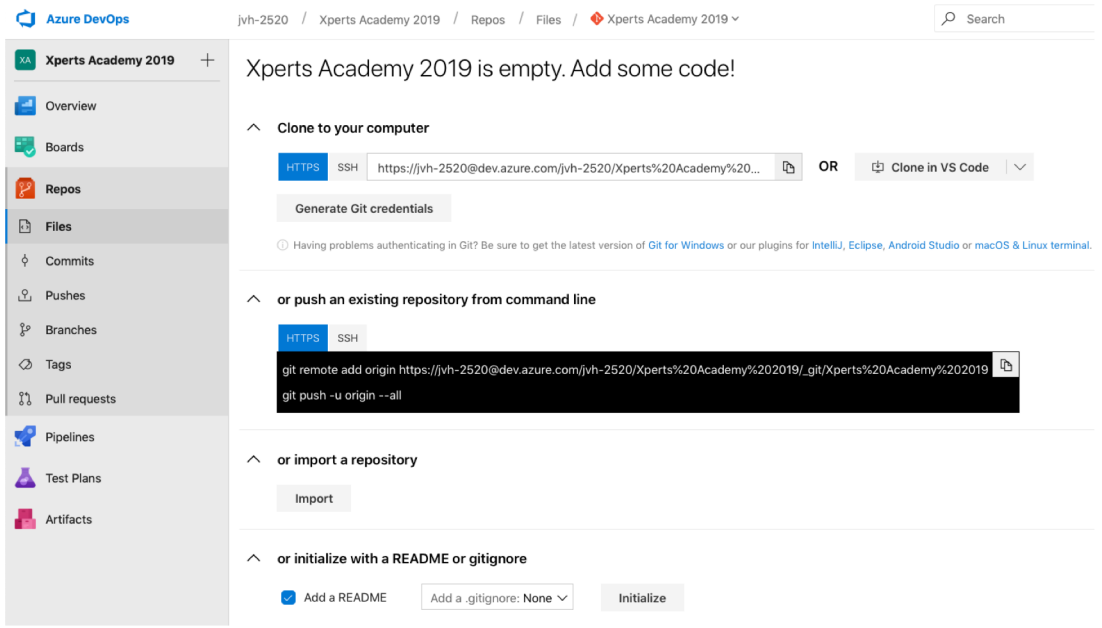




FORTINET NSE ACADEMY XPERTS

Import the ARM Template project

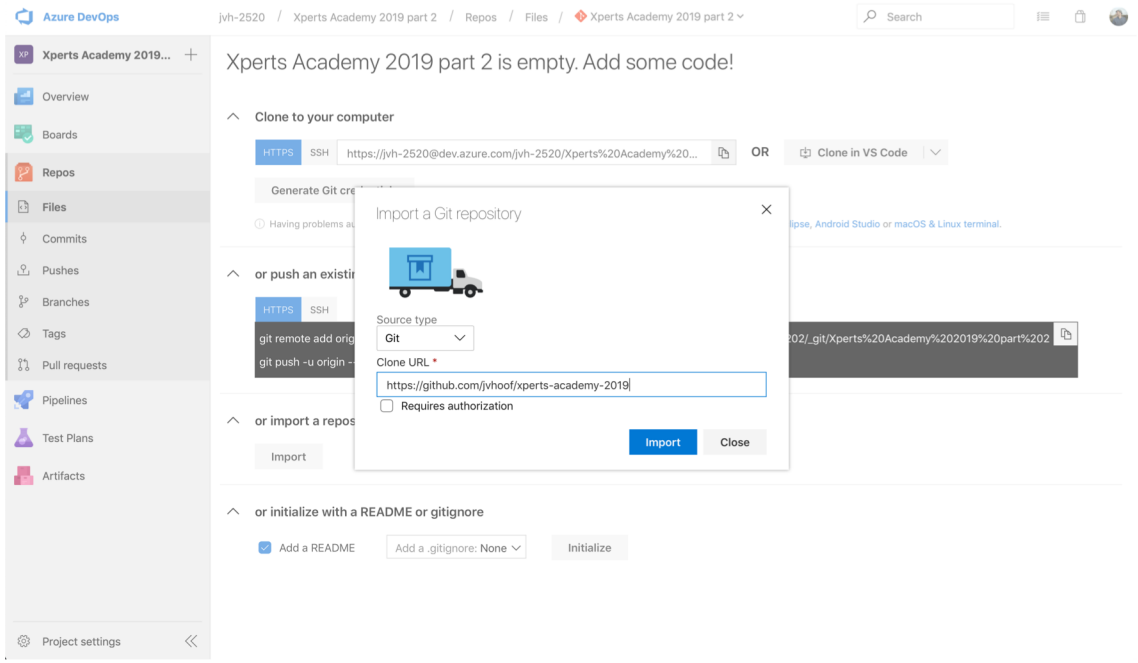
Go to Repos and select import

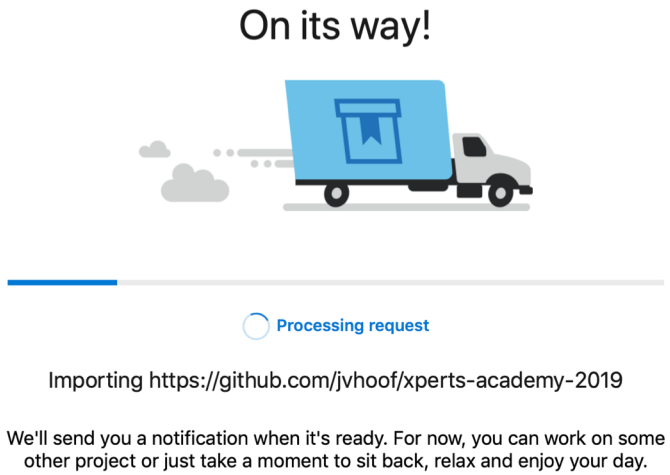


Import the ARM Template project

Import from the following Github project:

https://github.com/jvhoof/xperts-academy-2019.git



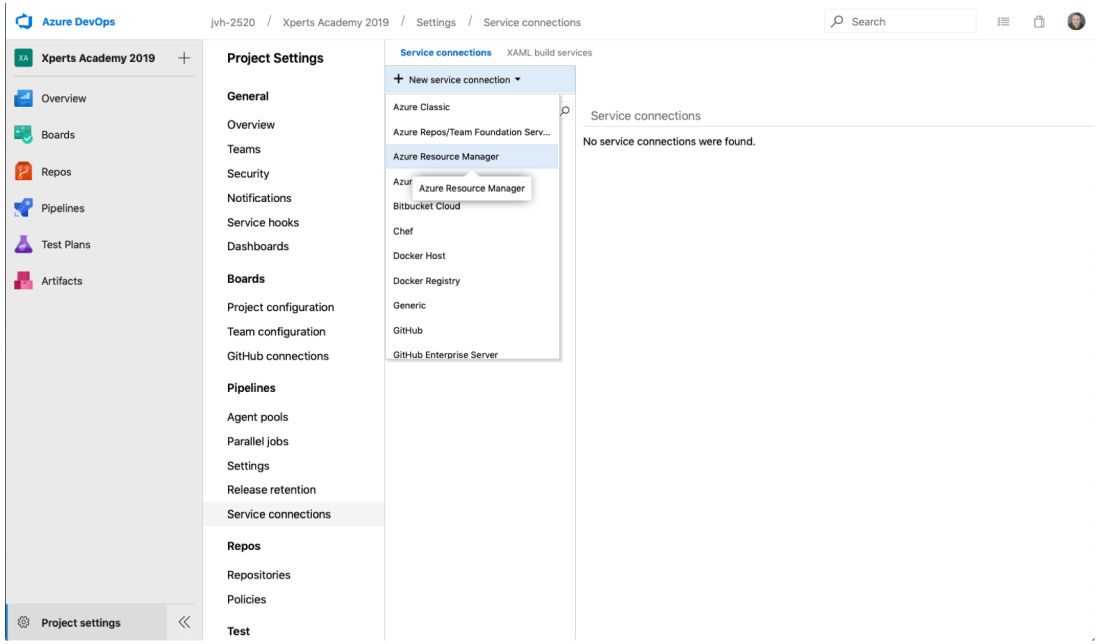


FORTINET NSE ACADEMY XPERTS

Connect Azure DevOps with Azure

Open the **Project settings** and look for **Service Connections**

Create a **New Service Connections** for Azure Resource Manager

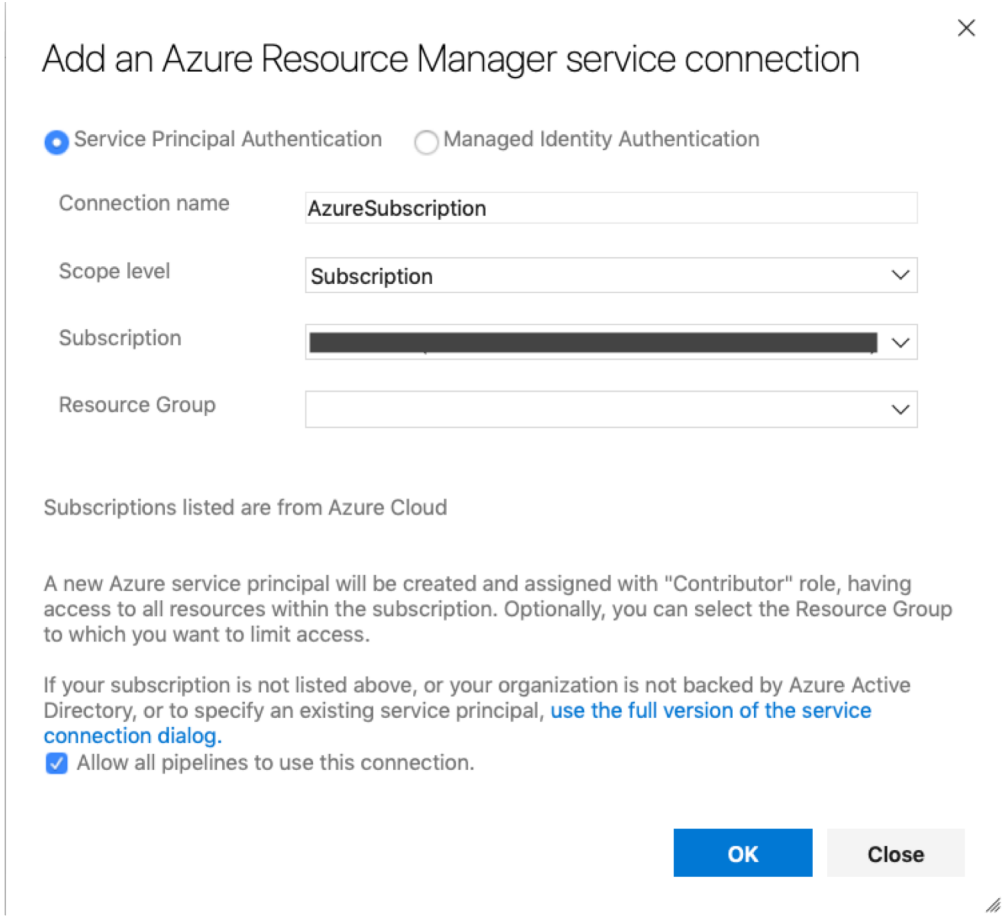


FORTINET NSE ACADEMY XPERTS

Connect Azure DevOps with Azure

Use the automatic way if you use the same credentials for Azure and Azure DevOps

Give the Connection the following name in one word and no quote's: "AzureSubscription"

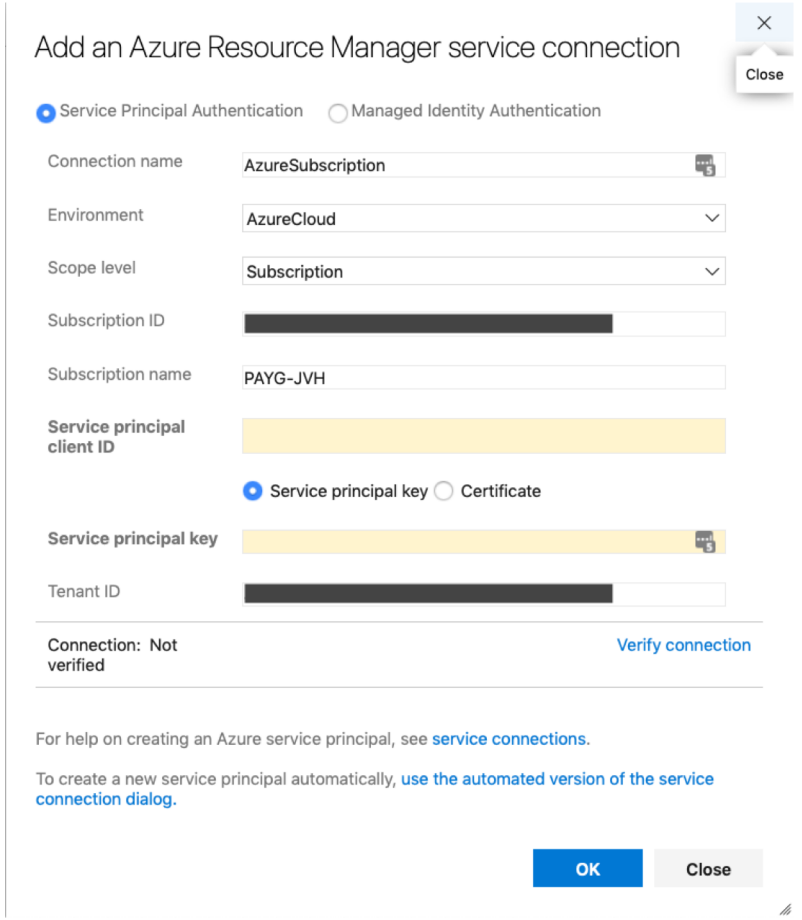


Connect Azure DevOps with Azure

Create a **service principal** or use **managed identity authentication** in case you have a different account

https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal

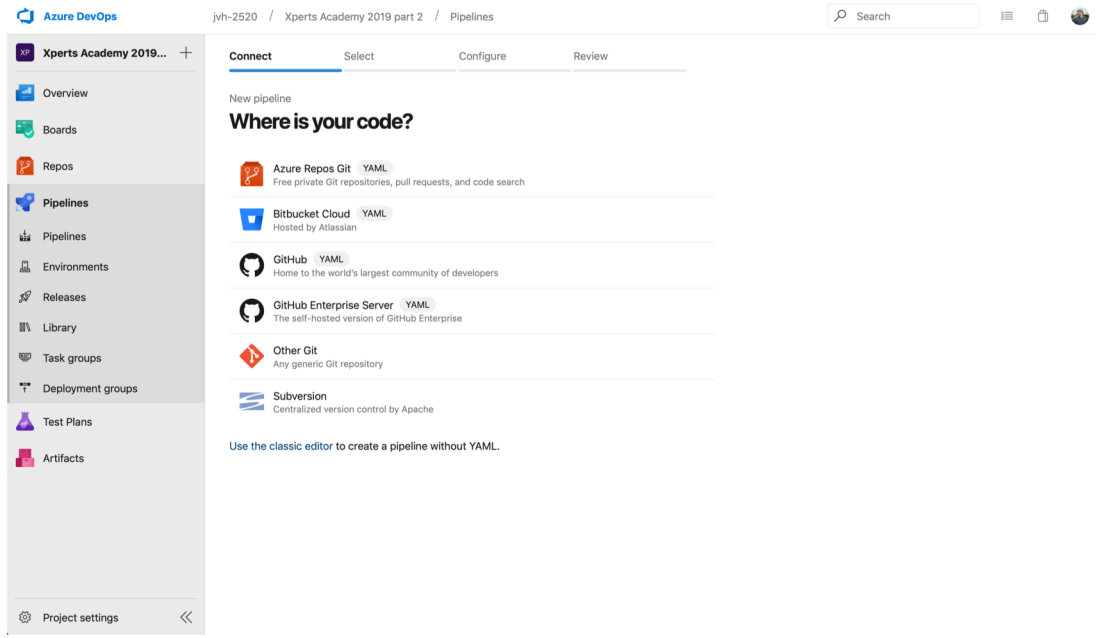
https://docs.microsoft.com/en-us/powershell/azure/create-azure-service-principal-azureps?view=azps-2.3.2



FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

Create a new build pipeline using the classic editor

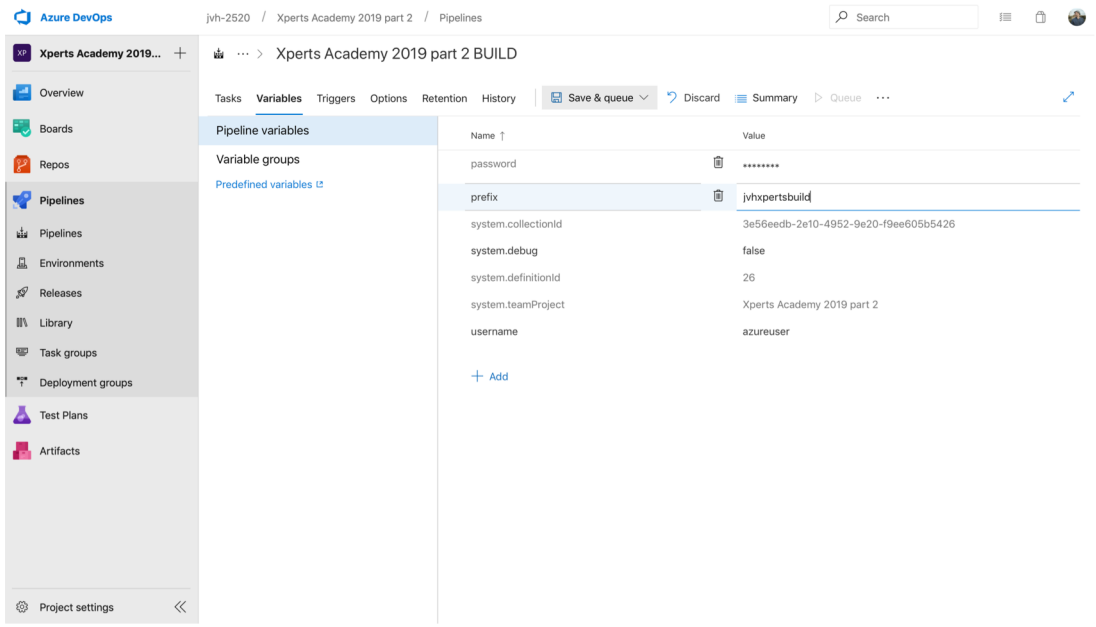


FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

Specify 3 variables and make sure to mask the password:

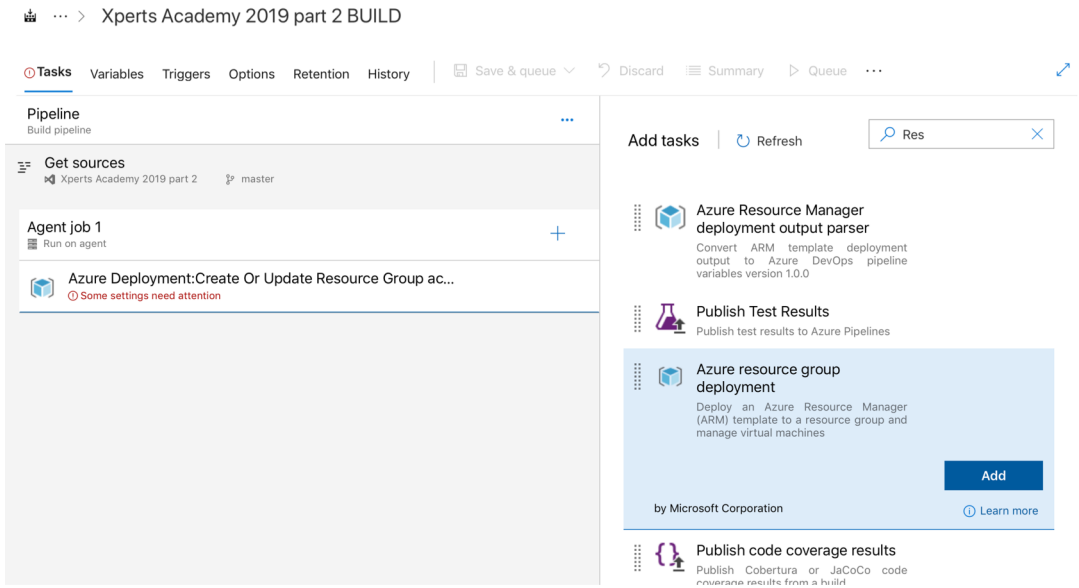
- username
- password
- prefix: max 12 characters, numbers and lowercase letters



FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

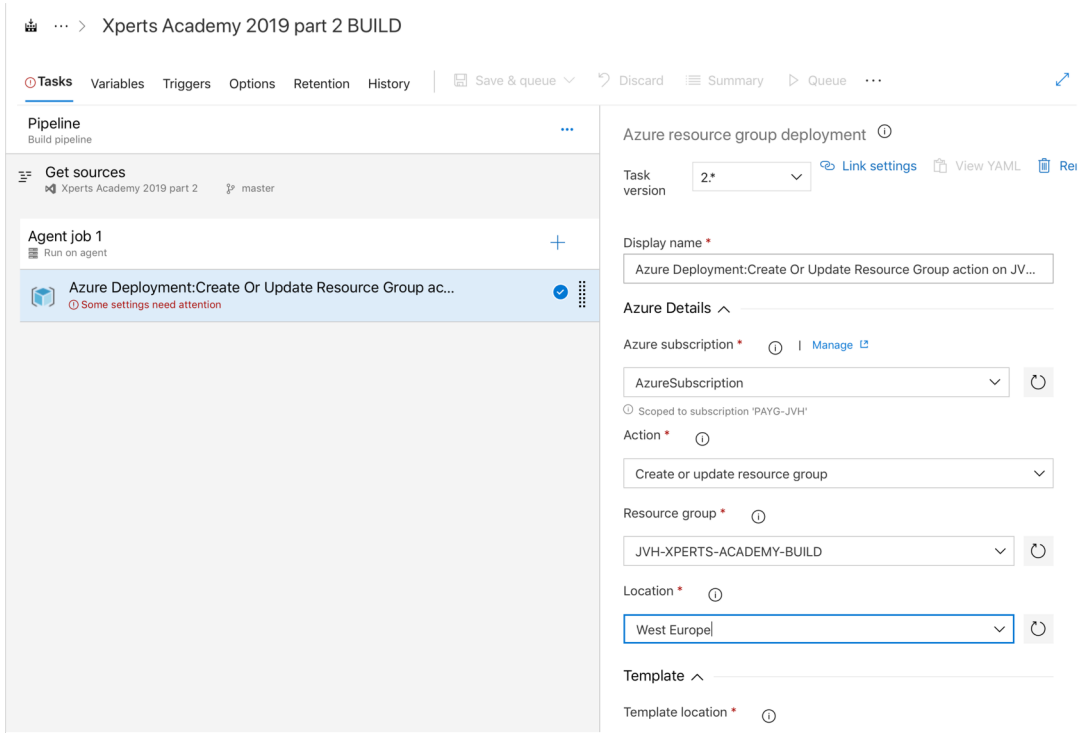
Add the first task: Azure Resource Group deployment



Create a new build pipeline

And configure it with the following variables:

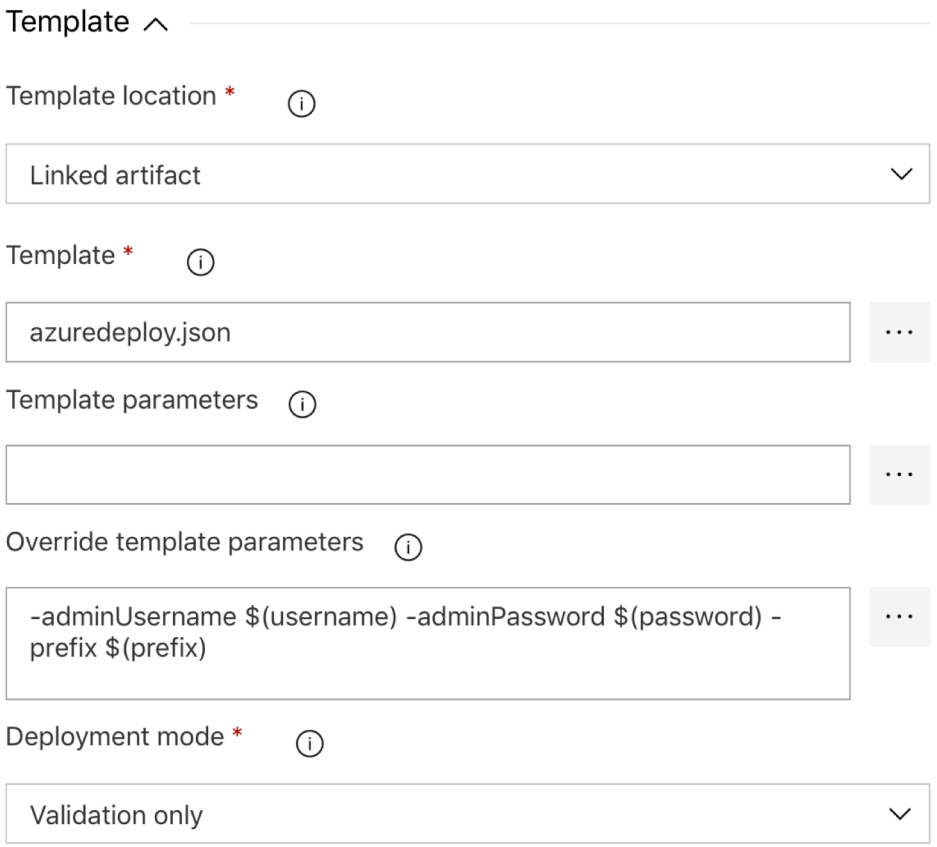
- Resource Group
- Location
- Template
- Template variables



Create a new build pipeline

And configure it with the following variables:

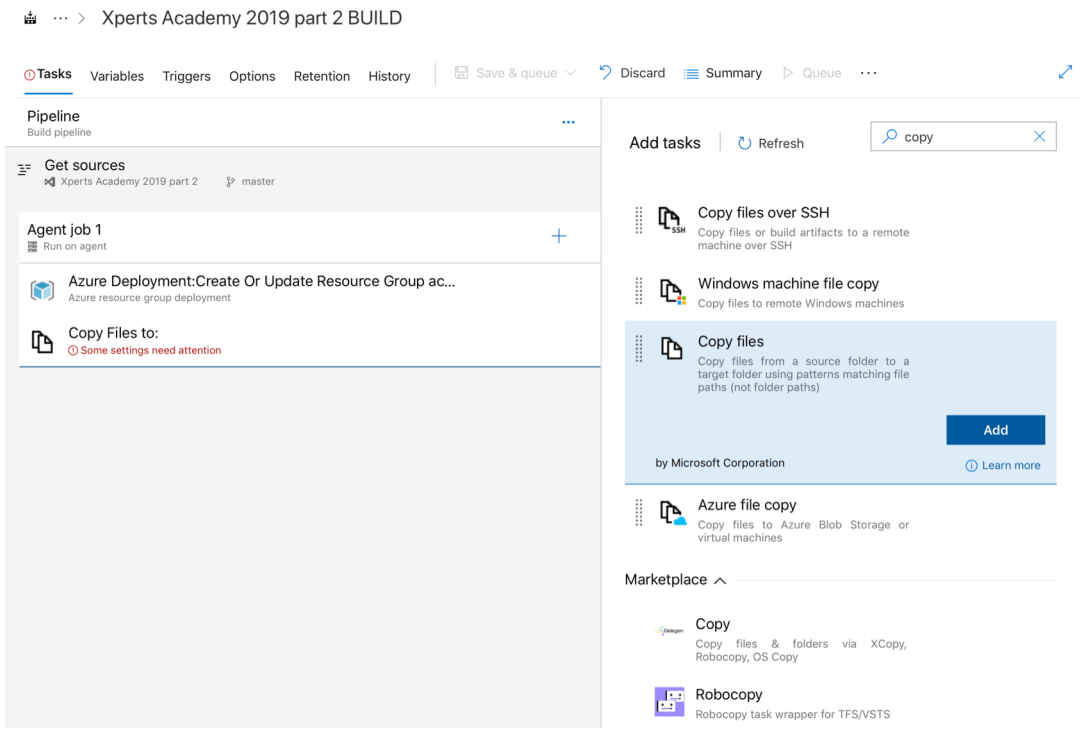
- Resource Group
- Location
- Template
- Template variables
- Deployment mode: validation only



FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

Add a copy task to copy the deployment files to the artifacts directory

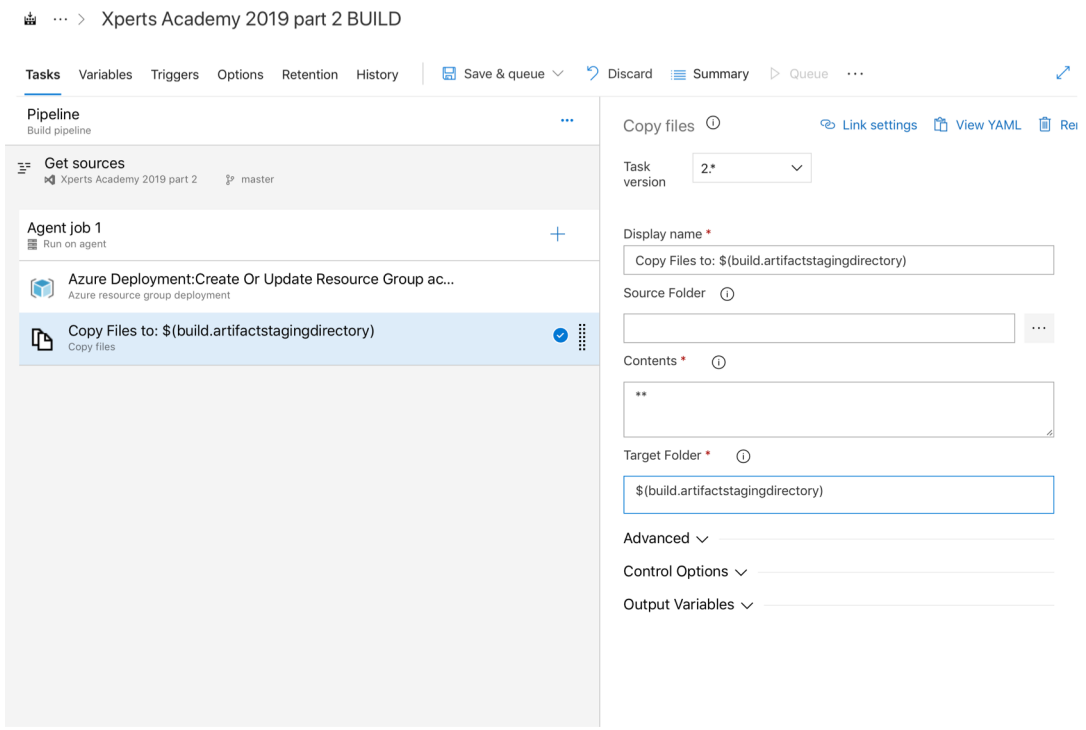


FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

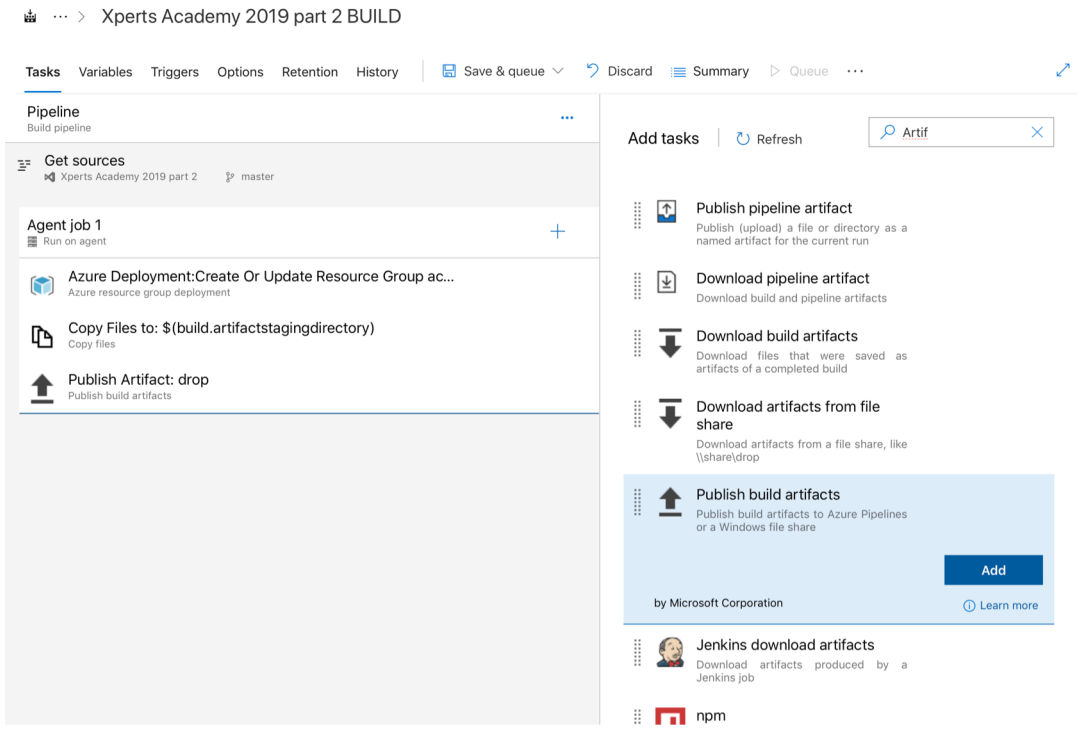
Add a copy task to copy the deployment files to the artifacts directory.

- Source: empty
- Target folder: $(build.artifactstagingdirectory)



Create a new build pipeline

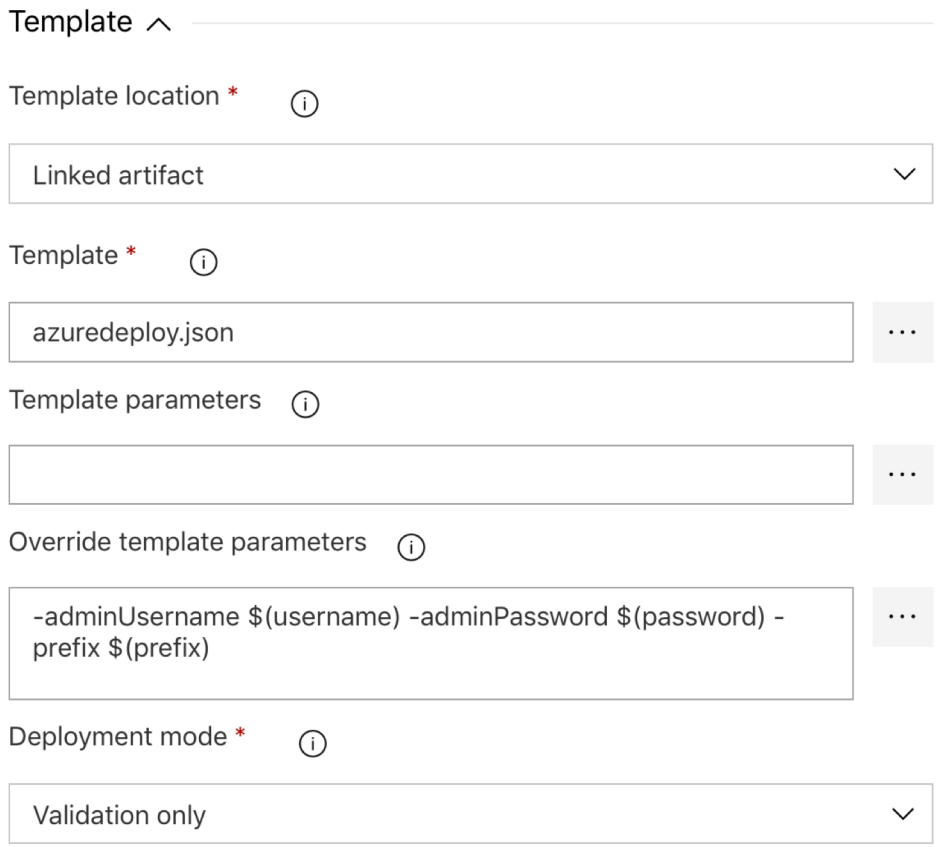
Add a publish build artifacts task



FORTINET NSE ACADEMY XPERTS

Create a new build pipeline

Add a publish build artifacts task

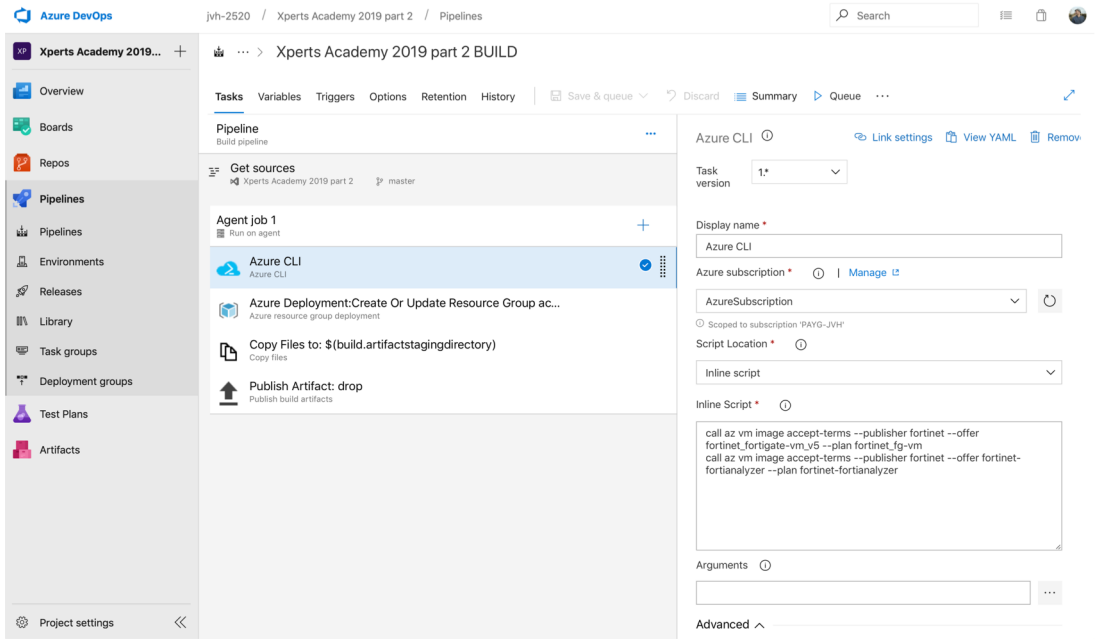


Create a new build pipeline

Never deployed a FortiGate or FortiAnalyzer?

Add an Azure CLI task to validate the EULA

This build will validate the ARM templates and create a package we can use in our release pipeline



FORTINET NSE ACADEMY XPERTS

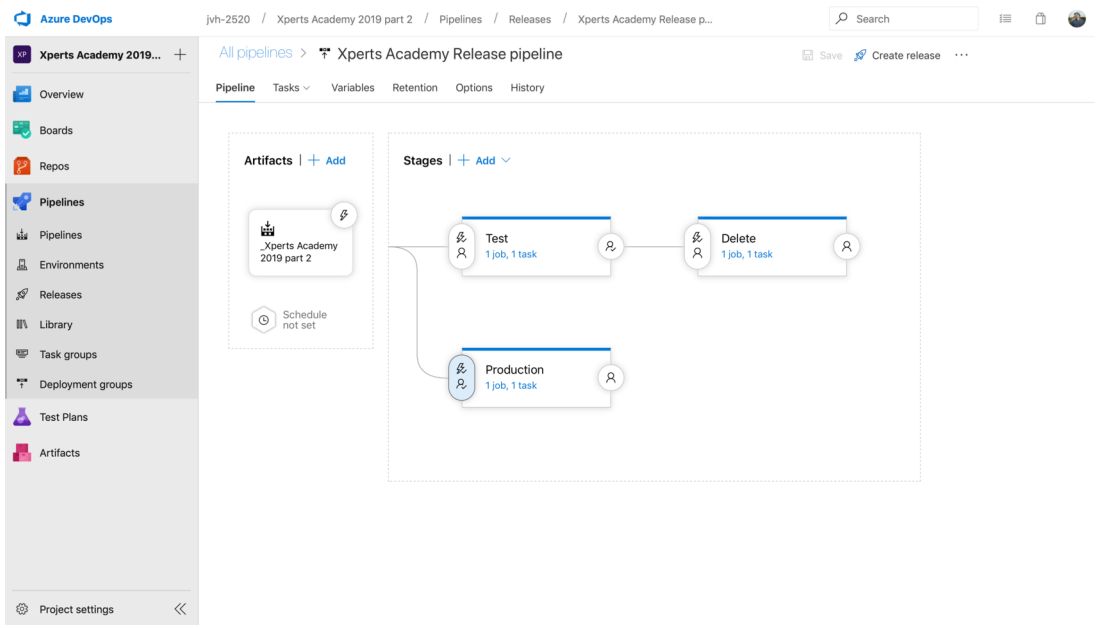
Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals



FORTINET NSE ACADEMY XPERTS

Infrastructure as Code

Step 3 : license file injection

Infrastructure as Code

Step 4 : Security Fabric deployment

FORTINET®
NSE ACADEMY
XPERTS